

Проектирование Data Warehouse (DWH) - основы

Модели данных

После определения ключевых слоев и подхода для вашего хранилища данных DWH, следующим шагом является выбор подходящей модели данных для каждого слоя. Этот выбор является важным, поскольку он определяет не только способ организации и хранения информации в хранилище, но и влияет на скорость разработки и удобство дальнейшей поддержки системы.

Выбор модели данных — это всегда поиск баланса. Если вы уделяете меньше времени на проектирование, хранилище может потребовать больше усилий для поддержки и наоборот. Поэтому важно найти золотую середину, которая будет удовлетворять текущим и будущим потребностям организации.

Каждый слой в DWH может быть построен на базе различных систем управления базами данных (СУБД), а также может использовать разные подходы к организации данных, например, реляционные, документо-ориентированные или колоночные модели. Выбор модели данных зависит от характера данных в каждом слое и задач, которые этот слой должен выполнять.

Например, для слоя staging может подойти более гибкая модель данных, так как здесь хранятся сырые, необработанные данные. В то время как для слоев, таких как ODS или DDS, где данные уже структурированы и интегрированы, может потребоваться более сложная и нормализованная реляционная модель данных.

Важно также учитывать, что разные модели данных могут влиять на производительность и способность системы выполнять сложные запросы, что особенно важно для слоев, предназначенных для аналитики и отчетности.

Какие модели бывают

1. Модель "Отсутствие модели" — это подход, когда структура данных не нормализована и все данные хранятся в одной таблице или нескольких плоских таблицах без явных отношений. Этот метод может быть применим в ситуациях, когда нужно очень быстро развернуть систему, как это часто бывает на ранних стадиях стартапов, где скорость ценится выше, чем строгая структура и долгосрочная масштабируемость.

Преимущества этого подхода включают:

- Быстрота: Поскольку нет сложных отношений, запросы выполняются быстрее, а разработка самой базы занимает меньше времени.
- Простота: Сама модель проста в реализации и управлении, так как не требует создания сложной схемы базы данных.

- Легкость запросов: Запросы к базе данных упрощаются, поскольку часто нет необходимости в операциях объединения (JOIN), что делает их более понятными и легкими для написания.

Недостатки могут включать:

- Дублирование данных: Все данные хранятся как есть, что может привести к избыточности и дублированию.
- Масштабирование: При увеличении объема данных и росте компании такая структура может стать неэффективной и трудной в обслуживании.
- Жесткость: Любые изменения в источниках данных могут потребовать значительных изменений в базе данных, что может быть трудоемким и рискованным.

"Отсутствие модели" может быть хорошим временным решением, когда нужно быстро запустить проект, но с ростом и развитием проекта может потребоваться переход к более структурированным и нормализованным моделям данных.

2. Модель "большие плоские данные", которую иногда называют "одна большая таблица" или "flat" модель, по сути, является методом хранения данных, при котором все записи содержатся в одной большой, расширенной таблице. Это напоминает огромный лист в программе Excel, где каждый столбец представляет определенный атрибут, а каждая строка — уникальную запись.

Преимущества такого подхода:

- Простота: Нет необходимости создавать сложные схемы баз данных с множественными таблицами и отношениями.
- Легкость доступа: Данные легко извлекаются, так как все находится в одной таблице.
- Удобство для пользователя: Пользователи, особенно те, кто привык работать с Excel, могут найти этот метод более интуитивно понятным.
- Быстрота разработки: Можно быстро начать работу с данными, так как нет необходимости в сложном проектировании и настройке.

Недостатки:

- Масштабируемость: По мере роста объема данных производительность может существенно снизиться из-за увеличения времени доступа и обработки.
- Избыточность: Возможно дублирование информации, так как одни и те же данные могут повторяться в разных строках.
- Сложность обновления: Обновление данных может стать громоздким, особенно если требуются изменения во многих местах.
- Управление изменениями: Любые изменения в структуре данных требуют ручной корректировки, что увеличивает риск ошибок.

- Аналитические ограничения: Сложность выполнения некоторых типов анализа из-за отсутствия нормализации и взаимосвязей между данными.

Такой подход может быть эффективен для небольших проектов или для начальных стадий, когда скорость и простота важны. Однако для крупных систем большие плоские таблицы могут привести к значительным проблемам в долгосрочной перспективе.

3. Модели "звезда" и "снежинка" — это две популярные схемы для организации данных в хранилищах данных (DWH), которые используются для облегчения процесса анализа и отчетности.

Star schema



Модель "звезда": Эта модель названа так из-за визуального сходства схемы со звездой, где в центре находится таблица фактов, окруженная таблицами измерений.

- Таблица фактов хранит количественные данные (например, продажи, цены, количество).
- Таблицы измерений содержат описательные данные (например, даты, клиенты, продукты) и связаны с таблицей фактов обычно через первичные ключи.

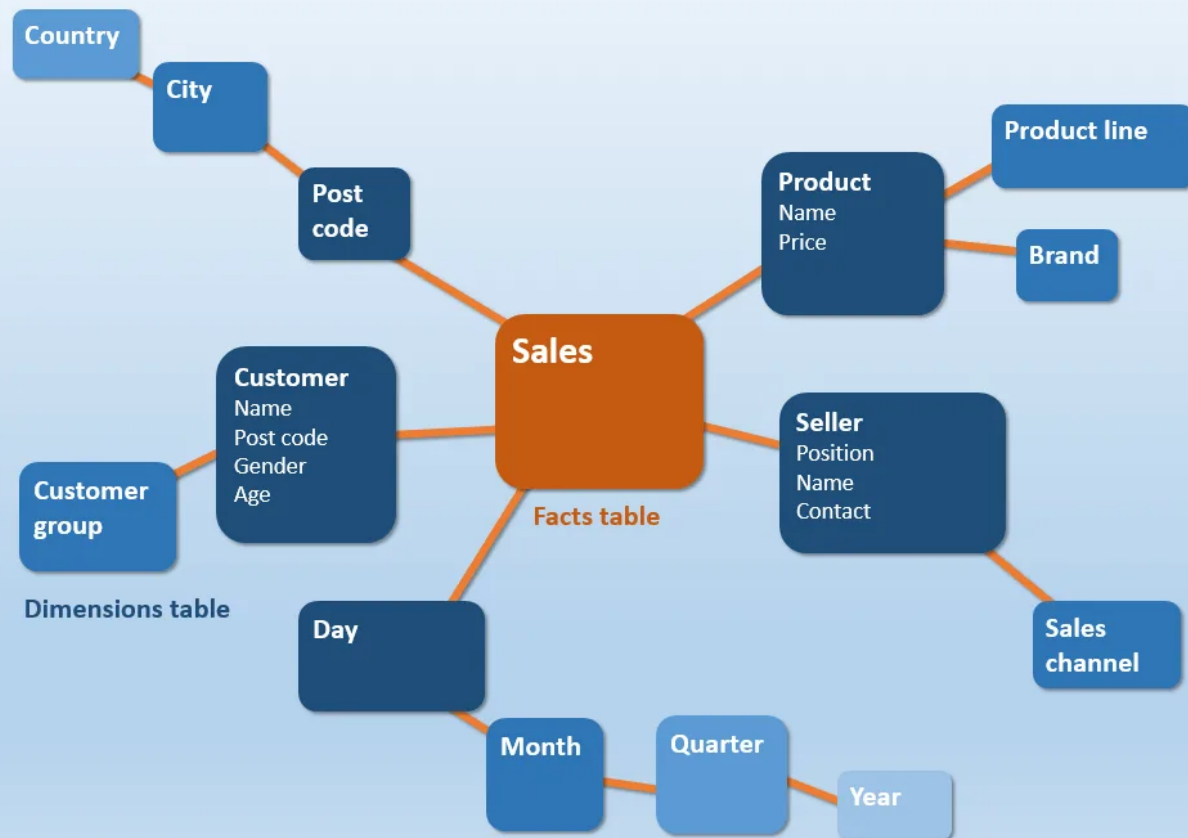
Преимущества:

- Простота понимания и использования.
- Хорошая производительность за счет прямых связей между фактами и измерениями.
- Эффективность для аналитических запросов, особенно при использовании OLAP-кубов.

Недостатки:

- Может привести к избыточности и повторению данных в таблицах измерений.
- Не идеально подходит для представления более сложных иерархий и отношений.

Snowflake schema



Модель "снежинка": Эта модель представляет собой расширение модели "звезда", где таблицы измерений нормализуются, разбиваясь на дополнительные таблицы, что создает схему, похожую на снежинку.

Преимущества:

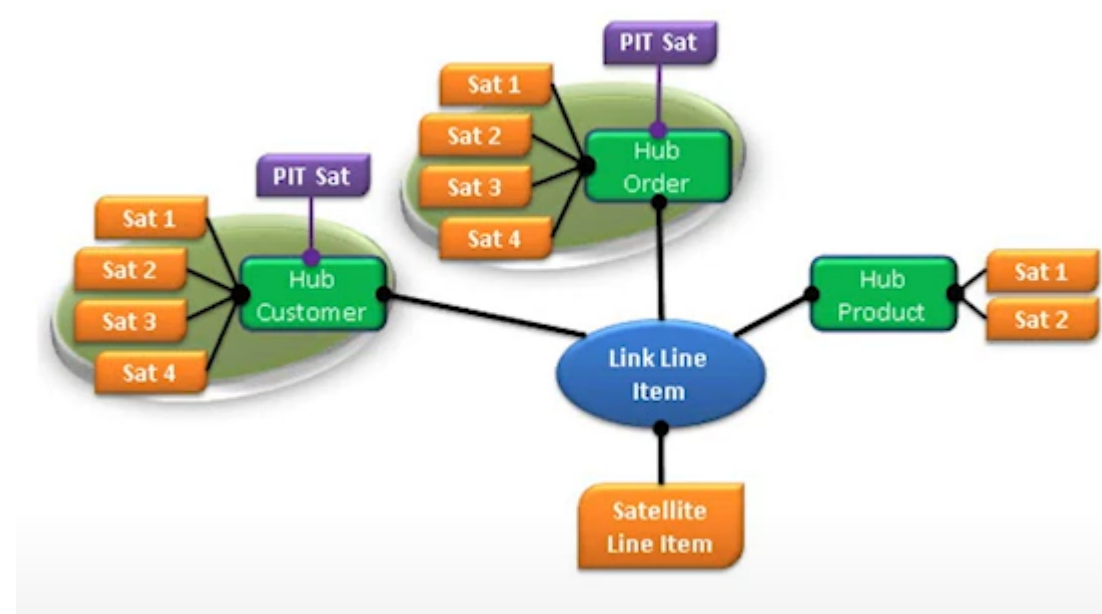
- Уменьшает избыточность данных за счет нормализации, что может уменьшить объем хранения.
- Лучше отображает сложные отношения и иерархии в данных.

Недостатки:

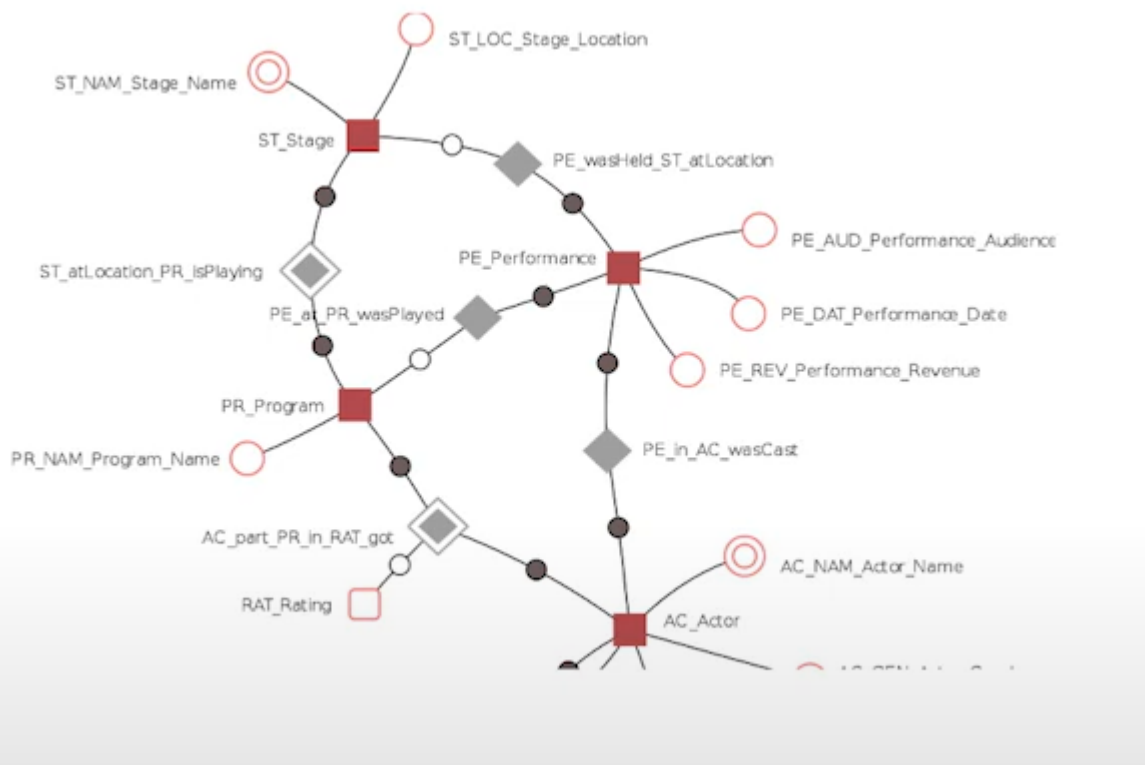
- Может быть более сложной для понимания и управления.
- Производительность запросов может ухудшиться из-за большого количества JOIN-операций.

4. Сложные современные модели данных

Архитектуры с высокой степенью нормализации, такие как Data Vault и Anchor Modeling, разработаны для решения некоторых проблем с устойчивостью к изменениям, с которыми сталкиваются более традиционные подходы типа "звезда" и "снежинка". Эти методологии уходят за пределы третьей нормальной формы (3НФ), стремясь к еще большей нормализации. Чем выше степень нормализации, тем меньше изменений требуется при внесении новых данных — вместо изменений происходит добавление новых элементов.



Data Vault представляет собой модель, которая использует концепцию хабов, линков и спутников для хранения данных, обеспечивая гибкость и масштабируемость. Она хорошо подходит для компаний, где данные часто меняются, и требуется возможность быстро адаптироваться к этим изменениям без необходимости полного перепроектирования хранилища.



Anchor Modeling — это еще более нормализованный подход, достигающий шестой нормальной формы (6НФ). Он разрабатывает строгую структуру с множеством различных специализированных типов таблиц, что обеспечивает высокую устойчивость к изменениям в бизнес-процессах.

Преимущества этих подходов:

- Устойчивость к изменениям: Модели легко адаптируются к новым требованиям бизнеса без необходимости перестроения всей структуры.
- Отсутствие дублирования: Данные не дублируются, что повышает качество и целостность информации.
- Гибкость: Можно легко добавлять новые данные и атрибуты без потери производительности.

Недостатки этих подходов:

- Сложность проектирования: Требуется глубокое понимание моделирования данных и специализированных знаний для создания и поддержки таких хранилищ.

- Сложность запросов: Может возникнуть необходимость использования большого количества JOIN-операций, что может уменьшить скорость выполнения запросов, особенно в системах, не оптимизированных для этого.
- Ресурсоемкость: Для поддержания и развития таких хранилищ требуется больше специалистов и времени.

Попробуем для каждого слоя выбрать модель данных (как один из вариантов):

1. Staging (STG): Здесь данные загружаются в их исходном виде. Лучше всего подходит гибкая модель данных, поскольку она позволяет быстро и без изменений переносить данные из источников. Это как принимать посылки в складском помещении без предварительной сортировки.
2. Operational Data Store (ODS): Для этого слоя целесообразно использовать реляционную модель данных, так как она позволяет эффективно управлять оперативными данными, обеспечивая точность и последовательность. Здесь данные уже должны быть более структурированными и организованными.
3. Detail Data Store (DDS): В этом слое предпочтительна нормализованная реляционная модель данных, так как DDS хранит исторические, детализированные данные. Это обеспечивает целостность и возможность глубокого анализа данных.
4. Common Data Marts (CDM): Здесь часто используется модель "звезда" или "снежинка" для реляционных баз данных, так как эти модели идеально подходят для аналитических запросов и отчетности. Они позволяют легко извлекать, агрегировать и анализировать данные.
5. Reporting Layer (REP): Для слоя отчетности подходит модель, которая оптимизирована для быстрого извлечения данных и их представления. Обычно это реляционная модель с оптимизацией для конкретных типов запросов и отчетов.